**Логика создания эффективных проверок**

Ранее уже было сказано: если у тест-кейса не указаны входные данные, условия выполнения и ожидаемые результаты, и/или не ясна цель тест-кейса — это плохой тест-кейс. И здесь во главе угла стоит **цель**. Если мы чётко понимаем, что и зачем мы делаем, мы или быстро находим всю остальную недостающую информацию, или столь же быстро формулируем правильные вопросы и адресуем их правильным людям.

Вся суть работы тестировщика в конечном итоге направлена на повышение качества (процессов, продуктов и т.д.). Но что такое качество? Да, существует сухое официальное определение (Quality. The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations. [ISTQB Glossary]), но даже там сказано про «user/customer needs and expectations» (потребности и ожидания пользователя/заказчика). И здесь проявляется главная мысль: качество — это некая ценность для конечного пользователя (заказчика).

Если мы подходим к тестированию формально, мы рискуем получить продукт, который по документам (метрикам и т.д.) выглядит идеально, но в реальности никому не нужен. Поскольку практически любой современный программный продукт представляет собой непростую систему, среди широкого множества её свойств и функций объективно есть самые важные, менее важные и совсем незначительные по важности для пользователей. Если усилия тестировщиков будут сконцентрированы на первой и второй категории (самом важном и чуть менее важном), наши шансы создать приложение, удовлетворяющее заказчика, резко увеличиваются. Есть простая логика:

• Тесты ищут ошибки.

• Но все ошибки найти невозможно.

• Значит, наша задача — найти максимум ВАЖНЫХ ошибок за имеющееся время.

Под важными ошибками здесь мы понимаем такие, которые приводят к нарушению важных для пользователя функций или свойств продукта. Функции и свойства разделены не случайно — безопасность, производительность, удобство и т.д. не относятся к функциям, но играют ничуть не менее важную роль в формировании удовлетворённости заказчика и конечных пользователей. Ситуация усугубляется следующими фактами:

• в силу множества экономических и технических причин мы не можем выполнить «все тесты, что придут нам в голову» (да ещё и многократно) — приходится тщательно выбирать, что и как мы будем тестировать, принимая во внимание только что упомянутую мысль: качество — это некая ценность для конечного пользователя (заказчика);

• никогда в реальной жизни (как бы мы ни старались) мы не получим «совершенного и идеального набора требований» — там всегда будет некоторое количество недоработок, и это тоже надо принимать во внимание.

Однако существует достаточно простой алгоритм, позволяющий нам создавать эффективные проверки даже в таких условиях. Приступая к продумыванию чек-листа, тест-кейса или набора тест-кейсов, задайте себе следующие вопросы и получите чёткие ответы:

• Что перед вами? Если вы не понимаете, что вам предстоит тестировать, вы не уйдёте дальше бездумных формальных проверок.

• Кому и зачем оно нужно (и насколько это важно)? Ответ на этот вопрос позволит вам быстро придумать несколько характерных сценариев использования того, что вы собираетесь тестировать.

• Как оно обычно используется? Это уже детализация сценариев и источник идей для позитивного тестирования (их удобно оформить в виде чек-листа).

• Как оно может сломаться, т.е. начать работать неверно? Это также детализация сценариев использования, но уже в контексте негативного тестирования (их тоже удобно оформить в виде чек-листа).

К этому алгоритму можно добавить ещё небольшой перечень универсальных рекомендаций, которые позволят вам проводить тестирование лучше:

• Начинайте как можно раньше — уже с момента появления первых требований можно заниматься их тестированием и улучшением, можно писать чеклисты и тест-кейсы, можно уточнять план тестирования, готовить тестовое окружение и т.д.

• Если вам предстоит тестировать что-то большое и сложное, разбивайте его на модули и подмодули, функциональность подвергайте функциональной декомпозиции — т.е. добейтесь такого уровня детализации, при котором вы можете без труда удержать в голове всю информацию об объекте тестирования.

• Обязательно пишите чек-листы. Если вам кажется, что вы сможете запомнить все идеи и потом легко их воспроизвести, вы ошибаетесь. Исключений не бывает.

• По мере создания чек-листов, тест-кейсов и т.д. прямо в текст вписывайте возникающие вопросы. Когда вопросов накопится достаточно, соберите их отдельно, уточните формулировки и обратитесь к тому, кто может дать ответы.

• Если используемое вами инструментальное средство позволяет использовать косметическое оформление текста — используйте (так текст будет лучше читаться), но старайтесь следовать общепринятым традициям и не раскрашивать каждое второе слово в свой цвет, шрифт, размер и т.д.

• Используйте технику беглого просмотра для получения отзыва от коллег и улучшения созданного вами документа.

• Планируйте время на улучшение тест-кейсов (исправление ошибок, доработку по факту изменения требований и т.д.).

• Начинайте проработку (и выполнение) тест-кейсов с простых позитивных проверок наиболее важной функциональности. Затем постепенно повышайте сложность проверок, помня не только о позитивных{82} , но и о негативных{82} проверках.

• Помните, что в основе тестирования лежит цель. Если вы не можете быстро и просто сформулировать цель созданного вами тест-кейса, вы создали плохой тест-кейс.

• Избегайте избыточных, дублирующих друг друга тест-кейсов. Минимизировать их количество вам помогут техники классов эквивалентности{94} , граничных условий{95} , доменного тестирования{95} .

• Если показательность тест-кейса можно увеличить, при этом не сильно изменив его сложность и не отклонившись от исходной цели, сделайте это.

• Помните, что очень многие тест-кейсы требуют отдельной подготовки, которую нужно описать в соответствующем поле тест-кейса.

• Несколько позитивных тест-кейсов{82} можно безбоязненно объединять, но объединение негативных тест-кейсов{82} почти всегда запрещено.

• Подумайте, как можно оптимизировать созданный вами тест-кейс (набор тест-кейсов и т.д.) так, чтобы снизить трудозатраты на его выполнение.

• Перед тем как отправлять финальную версию созданного вами документа, ещё раз перечитайте написанное (в доброй половине случаев найдёте опечатку или иную недоработку).